HARISHANKAR VISHWANATHAN

https://harishankarv.github.io harishankar.vishwanathan@gmail.com

My research focuses on building and verifying real-world systems using techniques from program verification, static analysis, abstract interpretation, and program synthesis. I work at the intersection of programming languages and operating systems and have experience with compiler infrastructure, SMT-based verification, and operating systems internals. My recent work develops static analyses to help secure the eBPF kernel extension mechanism in Linux and has contributed upstream patches improving the performance and security of the Linux kernel. Prior to research, I worked for three years as a software engineer building and maintaining data ingestion and analytics systems in a large-scale fintech environment.

EDUCATION

Rutgers University ***(transferred from UC Irvine)*** Jan 2019–July 2026 (Expected)
Ph.D. in Computer Science GPA 3.9/4.0

University at Buffalo Aug 2014–Feb 2016
M.S. in Computer Science and Engineering GPA 3.63/4.0

University of Mumbai Jul 2009–Jun 2013
B.E. in Electronics and Telecommunication Engineering First Class

EXPERIENCE

Research Assistant, Rutgers University ***(advised by Dr. Srinivas Narayana & Dr. Santosh Nagarakatte)*** Jan 2021–Present
Developing and improving static analyses for the Linux eBPF verifier

- Analyzing the Tristate Number (Bitfield) Abstract Domain (Tnums)
  - Analyzed the tristate number abstract domain used by the eBPF verifier to track different values that program variables in the eBPF bytecode can take across all executions.
  - Proved the soundness and optimal precision of existing algorithms for addition and subtraction using SMT solvers and manual proofs.
  - Developed a new, more precise algorithm for tristate number multiplication, which has been upstreamed to Linux.
- Verifying the Soundness of Multi-Domain Abstract Interpretation (Agni)
  - Analyzed the soundness of entire value-tracking analysis in eBPF, including the refinement logic between abstract domains (e.g. interval domain updating the tristate domain).
  - Wrote custom LLVM passes to automatically extract the semantics of the eBPF abstract operators from their C implementation (LLVM IR) into SMT logic.
  - Developed soundness conditions specifying the correctness of the abstract operators and verified the operators against the specification using SMT solvers.
  - Developed a counterexample-driven synthesizer to generate minimal eBPF programs when the soundness checks failed, exposing soundness bugs in the abstract operators, and reducing false positives from verification.
  - Maintained and helped build CI infrastructure running Agni daily against various Linux kernel trees (bpf/bpf-next/stable).
  - Contributed multiple patches to the eBPF verifier, fixing latent unsoundness and improving the precision of existing abstract operators, which have been upstreamed to Linux.
- Automatically synthesizing abstract operators for eBPF (Vayu)
  - Developed a framework to automatically synthesize abstract operators for eBPF combining techniques from component-based synthesis and abstract operator synthesis.
  - Synthesized more precise operators for interval addition and subtraction, which have been upstreamed to Linux.

Research Assistant, University of California, Irvine ***(advised by Dr. Anton Burtsev)*** Sep 2019–Dec 2020
Designing fast hash tables for k-mer counting

- Designed and implemented a cache-aware, linear-probing, partitioned hash table to count nucleotide sequences (k-mers), optimized for high-bandwidth DRAM systems.
- Developed a parallel parser for converting FASTQ files into a compact binary representation.
- Created a cross-core communication framework with SPSC queues using B-Queue, a fast lock-free queue, to insert the parsed k-mer sequences into partitioned hash tables in parallel.
- Optimized for performance by tweaking hashing schemes and buckets to prefetch.

Research Assistant, University at Buffalo ***(advised by Dr. Steven Y. Ko)*** Jan 2019–Jul 2019
Developing richer runtimes for ARM TrustZone trusted applications

- Conducted exploratory research on memory management in trusted operating systems, focusing on OP-TEE.
- Investigated runtime systems and interfaces for garbage collection algorithms, aiming to enable automated memory management for Trusted Applications in the Secure World by utilizing a Garbage Collector running partially in the Normal World.

Software Engineer, FactSet Research Systems Inc. Mar 2016–Jan 2019

Content Collection Services

- Contributed to building and maintaining the ETF Analytics platform, which provides detailed analytics on exchange-traded funds, including developing a suite of data collection tools, ingestion jobs, an Analytics Calculation Engine, and front-end applications for process and data quality control.
- Led the team's successful transition to a development workflow using Git and Jenkins for continuous integration (CI).

Content and Technology Solutions

- Contributed to building and maintaining the Open:Factset infrastructure at DataExchange, which integrates diverse vendor data sets and makes them accessible to clients.
- Developed the vendor-integration process by modeling vendor data, defining vendor-specific schemas and parsing methods.
- Implemented a REST service to support vendor self-integration with the goal of streamlining vender-integration.

SKILLS

Systems: Linux (eBPF), Android/AOSP, OP-TEE
Compilers & Verification: Clang/LLVM, SMT Solvers (z3, CVC5)
Languages: C, C++, Python, Java, SQL

SOFTWARE

- Agni, a static analysis framework for the Linux eBPF Verifier
- Sound algorithm for tristate number multiplication (upstreamed to Linux).
- Patch fixing a latent unsoundness bugs in the eBPF verifier (upstreamed to Linux).
- Vayu, a framework to synthesize precise abstract operators for the eBPF verifier.
- Patch improving the precision of abstract operators for addition, subtraction, and multiplication in the eBPF verifier (upstreamed to Linux).

PUBLICATIONS

1. Harishankar Vishwanathan, Matan Shachnai, Srinivas Narayana, and Santosh Nagarakatte, "Automatic Synthesis of Abstract Operators for eBPF", in Workshop on eBPF and Kernel Extensions **(eBPF)**, 2025.
2. Matan Shachnai, Harishankar Vishwanathan, Srinivas Narayana, and Santosh Nagarakatte, "Comparing the Precision of Abstract Operators in the eBPF Verifier using Differential Synthesis", in Static Analysis Symposium **(SAS)**, 2025.
3. Harishankar Vishwanathan, Matan Shachnai, Srinivas Narayana, and Santosh Nagarakatte, "Fixing Latent Unsound Abstract Operators in the eBPF Verifier of the Linux Kernel", in Static Analysis Symposium **(SAS)**, 2024. Also accepted for presentation at the Linux Plumbers Conference **(LPC)**, 2024.
4. Harishankar Vishwanathan, Matan Shachnai, Srinivas Narayana, and Santosh Nagarakatte, "Verifying the Verifier: eBPF Range Analysis Verification", in Computer Aided Verification **(CAV)**, 2023. Also accepted for presentation at the Linux Plumbers Conference **(LPC)**, 2023.
5. Harishankar Vishwanathan, Matan Shachnai, Srinivas Narayana, and Santosh Nagarakatte, "Sound, Precise, and Fast Abstract Interpretation with Tristate Numbers" in Code Generation and Optimization **(CGO)**, 2022.
6. Harishankar Vishwanathan, Chang Min Park, Sidharth Kumar Mishra, Karthik Dantu, Steven Y. Ko, and Lukasz Ziarek, "Poster: Partitioning Garbage Collection Between the Secure and Normal Worlds for Trusted Applications", in Mobile Systems, Applications, and Service **(MobiSys)**, 2019.

AWARDS

- **Distinguished Paper Award:** Code Generation and Optimization (CGO) 2022
- **Travel/Conference Grants**: MobiSys 2019, and SIGCOMM 2025

TEACHING

CS 416/518 Operating Systems, Rutgers University	Sep 2023 – Dec 2023
ICS 143A Operating Systems, University of California, Irvine	Sep 2020 – Dec 2020
ICS 238P Operating Systems, University of California, Irvine	Apr 2020 – Jun 2020
ICS 33 Intermediate Programming, University of California, Irvine	Jan 2020 – Mar 2020
CSE 486/586 Distributed Systems, University at Buffalo	Sep 2019 – Dec 2019